



20/05/2021

# Flight Club

Travail pratique individuel



BRIAN.GL

CFPT

## Table des matières

1	Table des versions .....	3
2	Introduction.....	3
3	Rappel de l'énoncé .....	4
3.1	Organisation .....	4
3.2	Livrables.....	4
3.3	Matériel et logiciels .....	4
3.3.1	Matériel et logiciels à disposition.....	4
3.3.2	Matériel utilisé.....	5
3.4	Description de l'application.....	5
4	Méthodologie .....	5
4.1	S'informer.....	6
4.2	Planifier.....	6
4.3	Décider .....	6
4.4	Réaliser .....	7
4.5	Contrôler .....	7
4.6	Évaluer .....	7
5	Planification .....	7
5.1	Product backlog.....	7
6	Analyse fonctionnelle .....	11
6.1	Navigation du site.....	11
6.2	Design général.....	12
6.2.1	Page d'accueil.....	13
6.2.2	Page des vols .....	14
6.2.3	Page du vol détaillé .....	15
6.2.4	Page de modification du vol .....	15
6.2.5	Page d'ajout de vol .....	15
6.2.6	Mes invitations .....	15
6.2.7	Créer un groupe.....	15
7	Fonctionnalités .....	15
7.1	Fonctionnalités du site .....	15
7.1.1	Utilisateurs anonymes.....	15
7.1.2	Utilisateurs connectés .....	15
7.2	Fonctionnalités majeures .....	16
7.2.1	Inscription.....	16

7.2.2	Connexion.....	17
7.2.3	Déconnexion.....	18
7.2.4	Modification du profil.....	18
7.2.5	Vols .....	19
7.2.6	Groupes .....	21
7.2.7	Téléchargement.....	24
7.2.8	Ajout d'images.....	25
7.2.9	Suppression d'images.....	25
7.3	Sécurité du carnet de vol.....	25
7.3.1	Carnet personnel .....	25
7.3.2	Carnet de groupe.....	27
7.3.3	La suppression d'images.....	28
8	Plannification.....	28
9	Généralités concernant l'implémentation .....	28
9.1	Base de données.....	29
9.1.1	Le modèle de données final .....	29
9.1.2	Dictionnaire de données .....	29
10	Librairies et outils externes .....	31
10.1	Git .....	31
10.2	Bootstrap.....	31
10.3	Bootstrap Studio.....	31
10.4	FlatIcon .....	31
10.5	Expression régulière .....	31
10.6	Katalon Recorder .....	32
10.7	Trello.....	32
10.8	TCPDF.....	32
10.9	Figma .....	32
11	Plan de tests et tests .....	32
11.1	Périmètre des tests .....	32
11.2	Équipement de test .....	32
11.2.1	Environnement .....	32
11.3	Scénarios de test .....	33
12	Conclusion .....	39
12.1	Difficultés rencontrées .....	39
12.2	Améliorations possibles.....	40

12.2.1	Architecture technique.....	40
12.2.2	Fonctionnalités .....	40
12.3	Bilan personnel.....	40
12.4	Remerciements .....	40
13	Annexes .....	41
13.1	Bibliographie.....	41
14	Glossaire .....	41
14.1	Termes techniques .....	41
14.1.1	User stories (ou récit utilisateur).....	41
14.1.2	Importation (Anglais : upload).....	41
14.1.3	Carnet de produit (Anglais: product backlog) .....	41
14.1.4	Formulaire GET .....	41
14.1.5	Input hidden .....	41
14.1.6	Framework.....	41
14.1.7	Sticky.....	41
14.1.8	Hash .....	41
14.1.9	Salt .....	42
15	Résumé du TPI Flight Club .....	42

## 1 Table des versions

N° de version	Auteur	Changements apportés
1.0	Brian GOLAY <brian.gl@eduge.ch>	Version finale du document pour le rendu du TPI

## 2 Introduction

Ce document est un rapport présentant différents aspects de la conception du projet Flight Club. Ce projet a été réalisé dans le cadre du Travail pratique individuel (TPI) durant la session de mai 2021. Il a pour but de valider mes compétences acquises pendant la formation Informaticien CFC dispensée à l'école d'informatique du CFPT au Petit-Lancy. Flight Club a pour but de permettre aux pilotes réels et virtuels d'enregistrer leurs vols, et de les partager.

### 3 Rappel de l'énoncé

Les informations suivantes sont extraites du cahier de charges du TPI.

#### 3.1 Organisation

Élève
Brian GOLAY <brian.gl@eduge.ch>

Maître d'apprentissage
Mathieu Michaël <edu-mathieum@eduge.ch>

Experts	
Borys Folomietow <borys@folomietow.ch>	Jacques Ortola <jacques.ortola@gmail.com>

#### 3.2 Livrables

- Planning
- Rapport au format PDF avec le code source en annexe
- FAQ présente sur le site avec des références dans le site sur des ancres de la FAQ
- Résumé du rapport du TPI
- Journal de travail – carnet de bord
- Script pour la création de la base de données
- Script pour l'initialisation des données (compte administrateur, permissions)

#### 3.3 Matériel et logiciels

##### 3.3.1 Matériel et logiciels à disposition

- 1 ordinateur avec Windows 10 et 2 écrans
- Suite Microsoft Office
- EasyPHP ou WAMP ou un serveur équivalent
- PHP
- MySql
- Bibliothèque pour un affichage responsive (par exemple bootstrap)

- Bibliothèque pour la génération de PDF en PHP (par exemple TCPDF)
- Bibliothèques utilitaires (par exemple jQuery)
- Repo Git (par exemple github ou gitlab)
- Outils pour des tests automatiques comme Katalon
- MySQL Workbench

### 3.3.2 Matériel utilisé

- 1 ordinateur avec Windows 10 et 2 écrans
- Suite Microsoft Office
- EasyPHP
- MySQL Workbench
- Visual Studio code
- Git (Github)
- Google chrome
- Bootstrap
- TCPDF

## 3.4 Description de l'application

L'application sera développée à l'aide des langages web suivants : PHP, HTML et CSS. Les fonctionnalités suivantes doivent être implémentées :

- Création d'un utilisateur
- Pour un utilisateur connecté
  - Enregistrer ses vols
  - Modifier ses vols enregistrés
  - Changer son mot de passe
  - Créer un groupe
  - Rejoindre un groupe
  - Télécharger ses vols au format .csv et .pdf

## 4 Méthodologie

Pour planifier mon projet de TPI, je me suis basé sur la **méthode en 6 étapes** qui nous a été enseignée au CFPT.

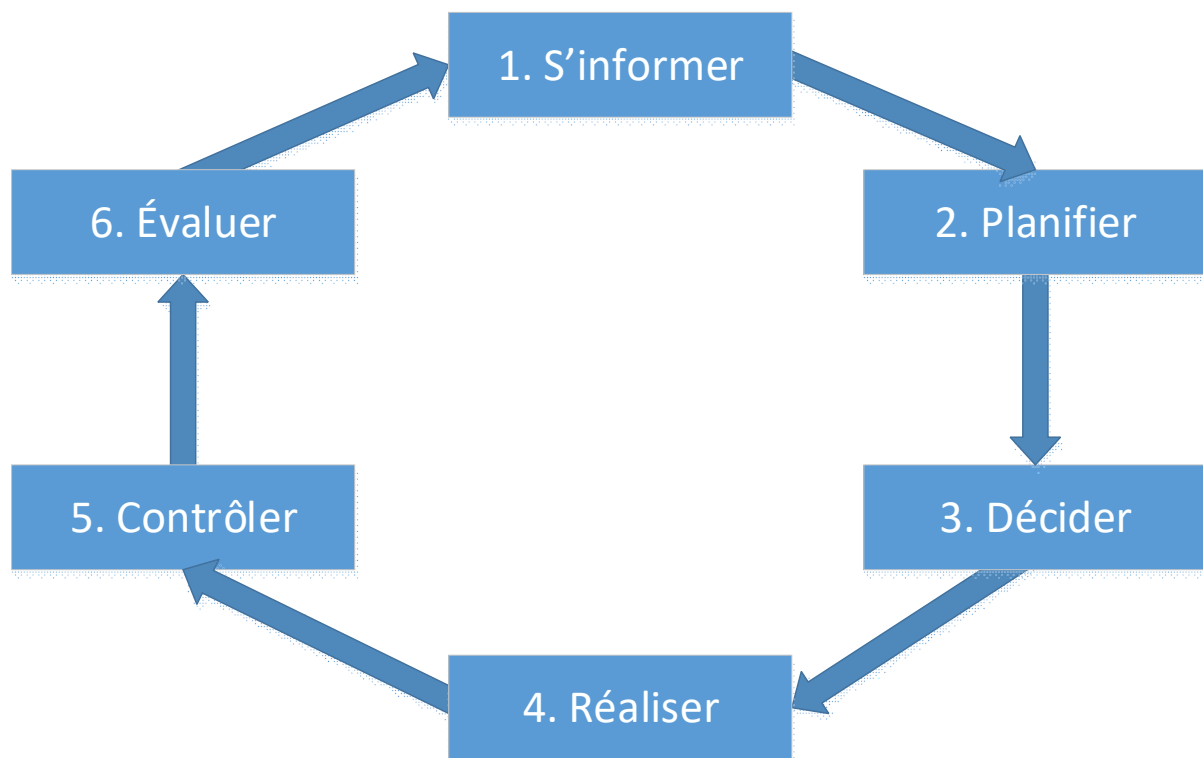


Fig. Méthode en 6 étapes

#### 4.1 S'informer

La toute première étape de mon projet était la lecture en profondeur de mon énoncé pour comprendre toutes les fonctionnalités qu'il était nécessaire d'implémenter. J'ai également demandé à mon formateur des clarifications sur des détails des spécifications lorsque c'était nécessaire.

#### 4.2 Planifier

Dès le début du projet, j'ai préparé un planning de travail pour savoir ce que j'avais besoin de faire et quand. Afin de pouvoir créer ce planning, j'ai eu besoin de découper mon travail en sous-tâches : j'ai donc divisé les divers points de l'énoncé sous forme d'user stories, qui présentent la tâche du point de vue de l'utilisateur final.

Pour chaque story, j'ai fixé une priorité pour m'aider ensuite à savoir ce qui doit être réalisé au plus vite (mes niveaux de priorité sont ! P0 : Bloquant, !! P1 : Critique, !!! P2 : Important et ? P3 : Secondaire).

Une fois les stories créées, je les ai stockées dans un product backlog. Ce fonctionnement est inspiré par la méthodologie Scrum (cependant, je n'ai pas appliqué entièrement cette méthodologie, car elle perd son utilité lorsqu'on travaille seul sur un projet).

J'ai ensuite créé un diagramme de Gantt, outil qui m'a permis de visualiser au quotidien ma progression dans le travail ainsi que les différences entre ma planification et mon avancement effectif.

#### 4.3 Décider

Au cours de l'avancement de mon travail, j'ai dû prendre de nombreuses décisions sur la manière de le réaliser. Lorsque je faisais des choix que je jugeais importants pour le projet, j'en ai parlé dans mon journal de bord en expliquant les raisons qui m'ont poussé à choisir ces options d'implémentation.

#### 4.4 Réaliser

Une fois les bonnes décisions prises, j'ai pu continuer sur l'implémentation de la fonctionnalité dans le code (ou la rédaction pour tout ce qui concerne la documentation).

#### 4.5 Contrôler

À chaque fois que je terminais une fonctionnalité, je l'ai testée dans différents cas d'usage pour être sûr qu'elle fonctionne comme prévu. J'ai documenté mes tests dans mon rapport TPI pour m'assurer qu'ils soient reproductibles par une personne externe au besoin.

J'ai aussi effectué à nouveau tous mes tests à la fin de chaque journée de travail pour éviter les régressions. J'ai ajouté à mon rapport TPI un tableau qui permet de visualiser l'avancement de l'état des tests au cours du projet.

Enfin, j'ai testé de fond en comble le programme à la fin de son implémentation sur d'autres plateformes (navigateurs et systèmes d'exploitation) pour m'assurer qu'il puisse bien fonctionner chez tout le monde.

#### 4.6 Évaluer

La dernière étape de la méthodologie en 6 étapes est l'évaluation, pour faire une rétrospective de ce que j'ai fait et avoir un regard critique sur ce qui pourrait être amélioré. Pour ce faire, j'ai écrit des bilans journaliers à chaque fin de journée de travail dans mon journal de bord. Pour avoir une auto-évaluation du travail en entier, j'ai également rédigé la partie Conclusion du rapport TPI.

### 5 Planification

#### 5.1 Product backlog

Nom	S1 : Utilisation d'un dépôt Git
Description (user story)	En tant que développeur, je peux utiliser Git sur le code source du projet. Je peux également accéder à un dépôt distant hébergé sur un site comme GitHub.
Critère d'acceptation	Un dépôt Git est configuré dans le dossier contenant le code source. Le code source du projet est accessible depuis le web à l'adresse indiquée dans la documentation.
Priorité	! P0 : Bloquant

Nom	S2 : Configuration de la base de données
Description (user story)	En tant que développeur, je peux utiliser une base de données MySQL respectant le modèle proposé par l'énoncé du TPI. Pour ce faire, j'ai accès à une classe côté PHP qui me permet d'effectuer les opérations de base (CRUD: create, read, update, delete) sur la base de données. Je souhaite également avoir dans le dépôt un dump de la structure de base de



	données qui me permettrait de la recréer sur un autre environnement.
Critère d'acceptation	Les tables Tbl_Flight, Tbl_Picture, Tbl_User-Flight, Tbl_User, Tbl_User-Group et Tbl_Group ont été créées et sont utilisables au moyen d'une classe Database
Priorité	! P0 : Critique

Nom	S3 : Affichage de la <i>home</i> page
Description (user story)	En tant qu'utilisateur non connecté, je ne peux pas encore accéder à mon profil : le site m'affiche une page présentant brièvement le projet et qui contient un bouton « Se connecter ».
Critère d'acceptation	Les tests concernant la fonction se connecter se passe sans problème.
Priorité	!! P1 : Critique

Nom	S4 : Enregistrements des vols
Description (user story)	En tant qu'utilisateur, je peux enregistrer mes vols
Critère d'acceptation	Les tests concernant l'enregistrement des vol pour un utilisateur connectés passent
Priorité	! P0 : Bloquant

Nom	S5 : Téléchargement du carnet de vol
Description (user story)	En tant qu'utilisateur, je peux télécharger au format CSV et PDF mes heures de vols
Critère d'acceptation	L'utilisateur télécharge son carnet de vol dans le format demandé
Priorité	!!! P2 : Important

Nom	S6 : Création de compte
Description (user story)	En tant qu'utilisateur non connecté, je veux pouvoir m'enregistrer pour accéder aux fonctionnalités des utilisateurs connectés
Critère d'acceptation	L'utilisateur peut se créer un compte
Priorité	! P0 : Bloquant

Nom	S7 : Connexion à son compte
Description (user story)	En tant qu'utilisateur non connecté, je veux pouvoir me connecter à mon compte pour accéder aux fonctionnalités des utilisateurs connectés
Critère d'acceptation	L'utilisateur se connecte avec succès
Priorité	! P0 : Bloquant

Nom	S8 : Modification des vols
Description (user story)	En tant qu'utilisateur connecté, je veux modifier mes vols
Critère d'acceptation	Modification du vol sélectionné avec succès dans la base de donnée
Priorité	!!! P2 : Important

Nom	S9 : Modification de mot de passe
Description (user story)	En tant qu'utilisateur connecté, je veux modifier mon mot de passe
Critère d'acceptation	Le mot de passe est modifier avec succès dans la base de donnée
Priorité	!! P1 : Critique

Nom	S10 : Créer un groupe
Description (user story)	En tant qu'utilisateur connecté, je veux créer un groupe
Critère d'acceptation	Un groupe est ajouté à la base de donnée avec l'utilisateur qui la créer
Priorité	! P0 Bloquant

Nom	S11 : Inviter des utilisateurs à mon groupe
Description (user story)	En tant qu'utilisateur connecté, je veux inviter un utilisateur dans mon groupe
Critère d'acceptation	Une invitation est créer dans la base de donnée
Priorité	!! P1 : Critique

Nom	S12 : Accepter une invitation à un groupe
Description (user story)	En tant qu'utilisateur connecté, je veux accepter une invitation pour rejoindre un groupe
Critère d'acceptation	Une invitation est modifiée dans la base de donnée
Priorité	!! P1 : Critique

Nom	S13 : Refuser une invitation à un groupe
Description (user story)	En tant qu'utilisateur connecté, je veux refuser une invitation pour rejoindre un groupe
Critère d'acceptation	L'invitation est supprimée dans la base de donnée
Priorité	!! P1 : Critique

Nom	S14 : Affichage des vols dans les groupes
Description (user story)	En tant qu'utilisateur connecté, je veux voir les vols des membres de mon groupe

Critère d'acceptation	Les vols des membres s'affichent
Priorité	!! P1 : Critique

Nom	S15 : Quitter un groupe
Description (user story)	En tant qu'utilisateur connecté, je veux quitter un groupe
Critère d'acceptation	L'utilisateur n'est plus dans le groupe dans la base de donnée et n'as plus accès aux informations du groupe
Priorité	! P0 : Bloquant

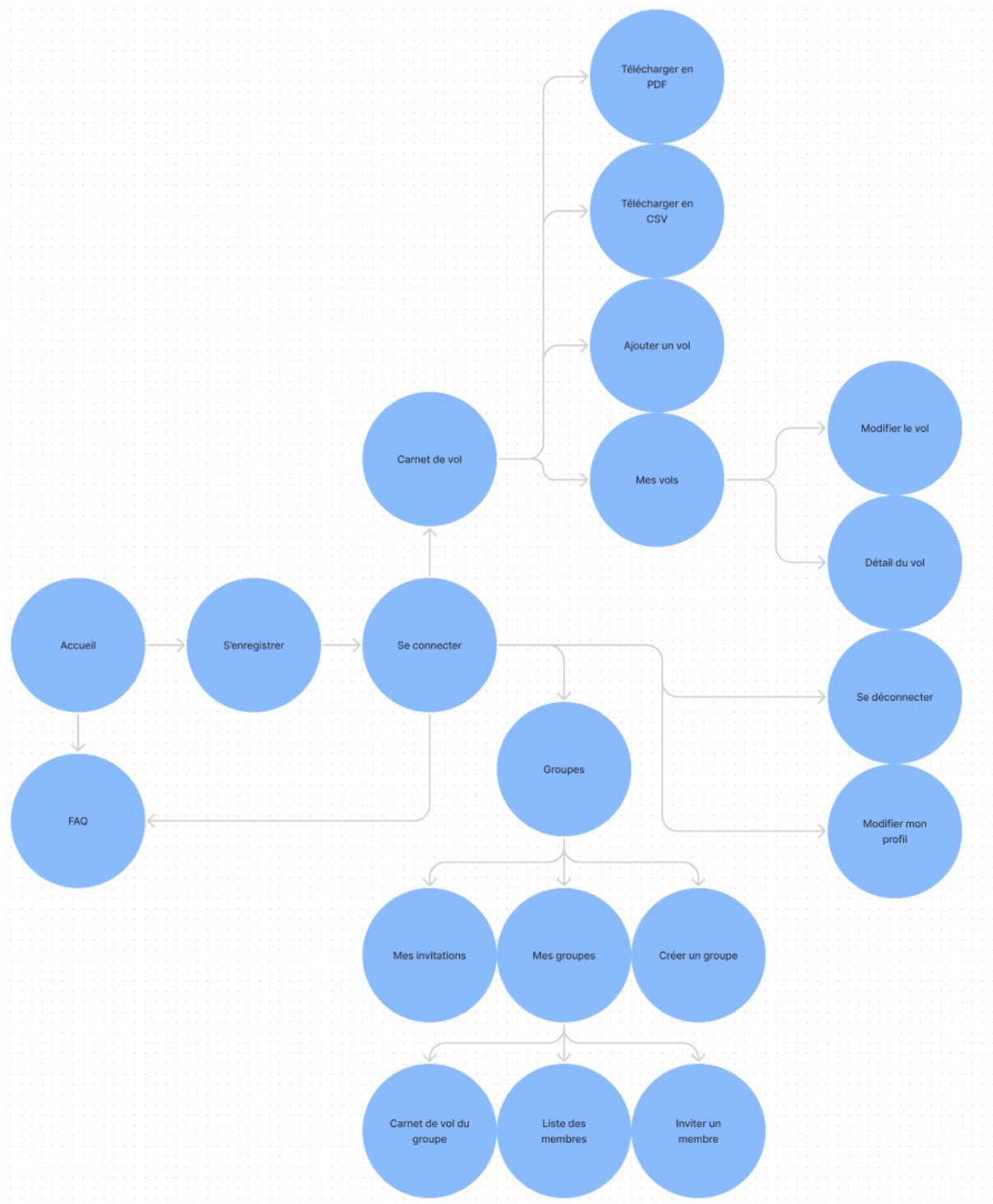
Nom	S16 : Accès à la FAQ
Description (user story)	En tant qu'utilisateur non connecté, j'ai la permission, comme tout le monde, de voir la page FAQ.
Critère d'acceptation	L'utilisateur n'est plus dans le groupe dans la base de donnée et n'as plus accès aux informations du groupe
Priorité	! P0 : Bloquant

! P0 : Bloquant, !! P1 : Critique, !!! P2 : Important et ? P3 : Secondaire

## 6 Analyse fonctionnelle

### 6.1 Navigation du site

Le site Flight Club contient 22 pages. Elles sont divisées en fonction des permissions de l'utilisateur, en plus de l'accueil et de la page FAQ. Il y a la partie utilisateur, la partie concernant les groupes et les vols.



## 6.2 Design général

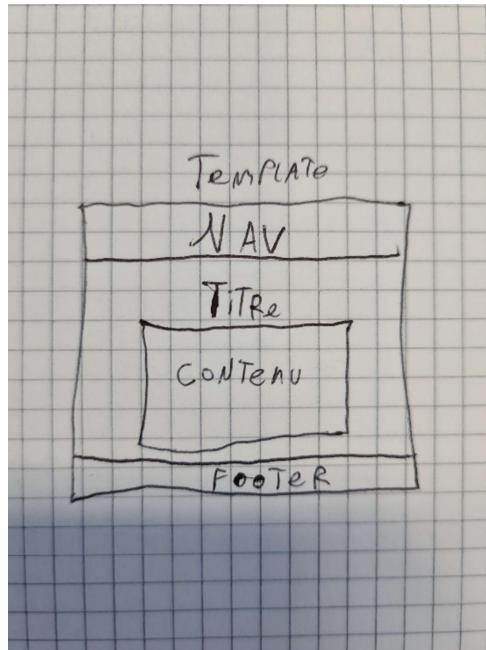


Fig. Structure des pages

Pour le design du site, j'ai choisi d'utiliser Bootstrap, car il s'agit d'un « *framework* » simple d'utilisation qui permet de créer une interface esthétique et compatible à tous les navigateurs. Toutes les pages du site ont la même structure pour faciliter la lecture et ne pas se perdre. Elle comprend un titre, une barre de navigation et son contenu. Le design du site est réalisé avec l'outil Bootstrap Studio (<https://bootstrapstudio.io/>)

## 6.2.1 Page d'accueil

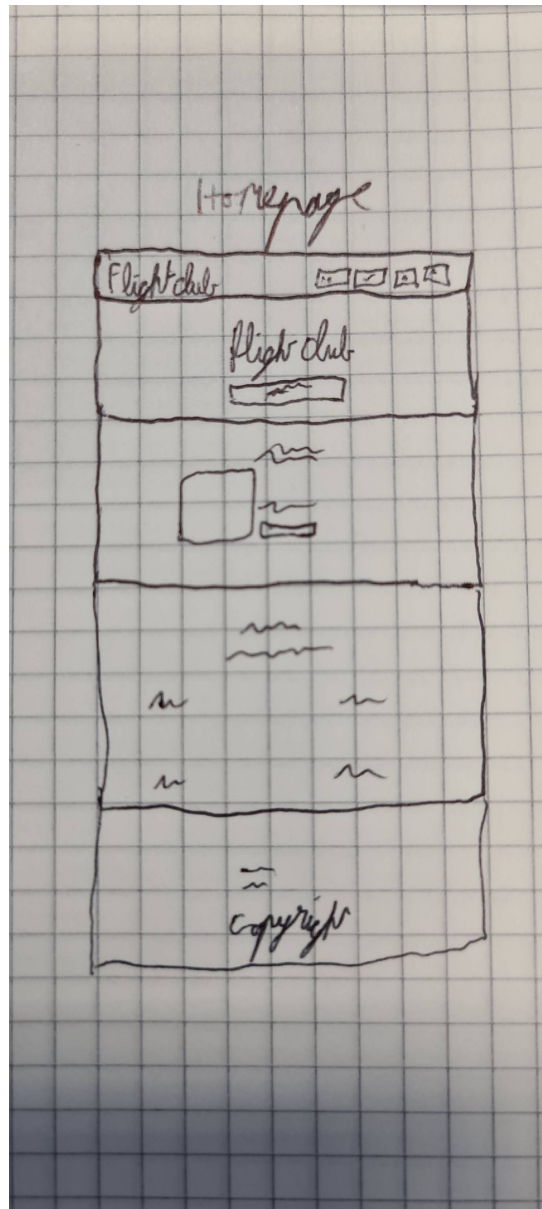


Fig. Page d'accueil

La page d'accueil est le point d'entrée du site. On peut y voir une courte description du site. De plus, la barre de navigation permet, en fonction de son niveau de permission, d'accéder aux autres pages du site.

## 6.2.2 Page des vols

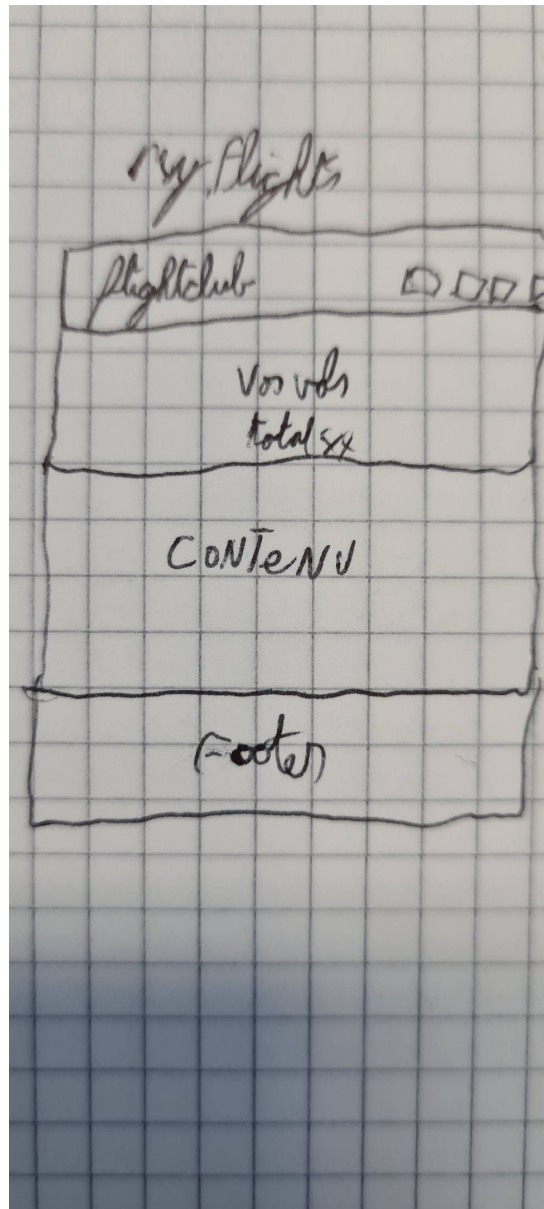


Fig. Page des vols

La page « myFlights » ou « page des vols » est la page où les vols de l'utilisateur s'affichent de manière simplifiée. L'utilisateur a la possibilité sur cette page d'avoir une vue détaillée de chacun de ses vols et d'éditer chaque vol. Il y a aussi un total d'heures de vol en haut juste sous le titre de la page.

### 6.2.3 Page du vol détaillé

La page du vol détaillé a un design similaire à la page des vols mais le contenu est moins large. Cette page contient toutes les informations saisies par l'utilisateur lors de la création du vol en plus des images importées. Les informations saisies sont triées par catégories sur la page.

### 6.2.4 Page de modification du vol

La page de modification du vol a un design identique à la page du vol détaillé à différence que les champs sont éditables. Les champs sont « *sticky* » et sont vérifié comme lors de l'ajout d'un vol. L'utilisateur a également la possibilité d'ajouter des images ou de les supprimer.

### 6.2.5 Page d'ajout de vol

La page d'ajout de vol a un design similaire aux design général. Cette page permet d'ajouter un vol avec une ou plusieurs images.

### 6.2.6 Mes invitations

La page « Mes invitations » est la page qui liste à l'utilisateur toutes les invitations qui lui ont été données. L'utilisateur a la possibilité de les accepter ou de les refuser. Le design est similaire à la page d'ajout de vol.

### 6.2.7 Créer un groupe

La page « Créer un groupe » est une page qui permet à tout utilisateur connecté de créer un groupe. Le design est similaire à la page d'ajout de vol.

## 7 Fonctionnalités

### 7.1 Fonctionnalités du site

La fonctionnalité principale de mon site est de sauver ses vols dans un carnet disponible partout dans le monde et de pouvoir télécharger ses heures de vols en .pdf ou en .csv facilement.

#### 7.1.1 Utilisateurs anonymes

C'est le niveau de permission le plus bas, ils représentent tous les utilisateurs du site qui ne sont pas connectés. Leurs fonctionnalités sont décrites dans le cahier des charges, et sont les suivantes :

- Accès à la page home
- Accès à la page FAQ
- Créer un compte
- Se connecter

#### 7.1.2 Utilisateurs connectés



Ce sont des utilisateurs lambda qui ont juste créé un compte sur le site. Ils sont définis par une adresse mail. Les fonctionnalités sont les suivantes :

- Modifier leur profil
- Enregistrer un vol
- Créer un groupe
- Inviter des autres utilisateurs dans leurs groupes
- Voir les heures de vols des groupes
- Quitter leurs groupes
- Se déconnecter
- Accéder à la page FAQ

## 7.2 Fonctionnalités majeures

### 7.2.1 Inscription

Cela permet d'avoir accès aux fonctionnalités avancées du site, comme les groupes, ou l'ajout de nouveaux vols. Les utilisateurs doivent entrer leur email et un mot de passe pour pouvoir se connecter. Lorsque l'utilisateur appuiera sur le bouton « S'enregistrer », la partie serveur va vérifier le contenu des champs, et retourner les erreurs commises à l'utilisateur. Dans le cas où aucune erreur n'est retenue, l'utilisateur est ajouté dans la base de données. La partie du code pour le mot de passe est une partie que j'ai inventé. Elle se passe comme cela: je commence par le « *hasher* » en sha512, ensuite on ajoute un « *salt* » qui est fait en prenant quelques parties du « *hash* » et en les mettant au début et à la fin. Finalement, on le « *hash* » encore une fois en sha512. Après ma partie, je mets ce « *hash* » dans la fonction « *password\_hash* » de PHP.

```
//if all data are not empty
if (!empty($email) && !empty($password) && !empty($password2)) {
    if (strlen($email) < 100) {
        //we check that the email is not already in our database
        if (userDAO::readEmail($email) == null) {
            //we check both of the password are the same and the strength of
            the password
            if ($password === $password2) {
                if (password_strength($password)) {
                    //we hash the password
                    $hashed = hash('sha512', $password);
                    $salted = substr($hashed, 0, 20) . $hashed . substr($hashed, 50, 70);

                    $hashed = hash('sha512', $salted);
                    //we hash again with password hash
                    $hashed = password_hash($hashed, PASSWORD_BCRYPT);
                    //we add it to the database
```

```
        userDAO::AddUsers($email, $hashed);
        $_SESSION['user'] = userDAO::readUsersByEmail($email);
        //we redirect him to the login page
        header("Location: ./index.php?page=login");
        exit();
    } else {
        $message = "Le mot de passe doit contenir au moins une maj
        uscule, une minuscule, un chiffre et un caractère spécial et faire 8 caractère
        ou plus!";
    }
    } else {
        $message = "Les mots de passes ne sont pas identiques !";
    }
    } else {
        $message = "L'email existe déjà";
    }
} else {
    $message = "L'email est trop long";
}
}
```

### 7.2.2 Connexion

Pour cette fonctionnalité, le serveur va d'abord valider les entrées utilisateurs avant de vérifier si l'utilisateur possédant ces identifiants existe avant de finalement sauvegarder les données de connexion dans la session.

```
//we check that the email and password are filled
if (!empty($email) && !empty($password)) {

    //we hash the user password
    $hashed = hash('sha512', $password);
    $salted = substr($hashed, 0, 20) . $hashed . substr($hashed, 50, 70);
    $hashed = hash('sha512', $salted);

    //we compare the hashed password of the user with the one in the database
    if (password_verify($hashed, userDAO::getUserByEmail($email)['Txt_Password
    _Hash'])) {
        //we connect the user in session
        $_SESSION['connected'] = true;
        $_SESSION['userID'] = userDAO::getUserByEmail($email)['Id_User'];
        $message = "Connecté avec succès";
        //we redirect the user to the homepage
        header("Location: ./index.php?page=homepage");
        exit();
    } else {
        $message = "Mauvais mot de passe ou email !";
    }
}
```

### 7.2.3 Déconnexion

La page de déconnexion est simple. Elle supprime la session et tous les cookies utilisés par le site et redirige l'utilisateur vers la page d'accueil.

```
//start a session
session_start();
$_SESSION = array();
//remove all cookies and destroy the session
if (ini_get("session.use_cookies")) {
    setcookie(session_name(), '', 0);
}
session_destroy();
//redirect the user to the homepage
header("Location: ./index.php?page=homepage");
exit;
```

### 7.2.4 Modification du profil

Cette fonctionnalité est accessible uniquement pour soi-même, seul l'utilisateur peut modifier son compte. L'informations pouvant être modifiée est son mot de passe. Si une erreur survient, l'utilisateur est averti avec un message d'erreur.

```
//we check that the email and password are filled
if (!empty($newPassword) && !empty($repeatNewPassword) && !empty($oldPassword)
) {

    //we hash the user password
    $hashed = hash('sha512', $oldPassword);
    $salted = substr($hashed, 0, 20) . $hashed . substr($hashed, 50, 70);
    $hashed = hash('sha512', $salted);

    //we compare the hashed password of the user with the one in the database
    if (password_verify($hashed, userDao::getUserByEmail(userDAO::getUserByID(
$_SESSION['userID'])(['Txt_Email'])(['Txt_Password_Hash']))) {
        //we check that booth of the new password are the same
        if ($newPassword == $repeatNewPassword) {
            //we hash the new password to make it work with the login
            $hashed = hash('sha512', $newPassword);
            $salted = substr($hashed, 0, 20) . $hashed . substr($hashed, 50, 70);

            $hashed = hash('sha512', $salted);

            //we hash again with password hash
            $hashed = password_hash($hashed, PASSWORD_BCRYPT);

            //we update the password in the database
```

```
        UserDAO::changePassword($_SESSION['userID'], $hashed);
        //we display a success message to the user
        $messageSucces = "Mot de passe changé avec succès";
    }else {
        //we display an error message to the user
        $messageErreur = "Les 2 mot de passe saisis ne sont pas identiques!";
    }
} else {
    //we display an error message to the user
    $messageErreur = "Mauvais mot de passe!";
}
}
```

## 7.2.5 Vols

### 7.2.5.1 Enregistrer un vol

Cette fonctionnalité est accessible uniquement un utilisateur connecté. Les données du vol saisies par l'utilisateur sont enregistrer dans la base de données si elles sont valides. Un message de succès ou d'erreur s'affiche en fonction du résultat.

```
//we check that every data is ok
if (empty($message)) {

    //we check with is the latest id in the database
    $allflight = FlightDAO::getAllFlightOrderById();
    $idFlight = 0;
    if (!empty($allflight)) {
        $idFlight = $allflight[0]['Id_Flight'] + 1;
    } else {
        $idFlight = 0;
    }
    try {

        //converting departure datetime to a good datetime
        $dateDeparture = explode("T", $dateDeparture);
        $dateDeparture = $dateDeparture[0] . " " . $dateDeparture[1];

        //converting arrival datetime to a good datetime
        $dateArrival = explode("T", $dateArrival);
        $dateArrival = $dateArrival[0] . " " . $dateArrival[1];

        //we add the flight
        FlightDAO::addFlight(
            $idFlight,
            $_SESSION['userID'],
            $role,
            strtoupper($flight),
```

```

        $dateDeparture,
        $dateArrival,
        $timeEngineOn,
        $timeEngineOff,
        $typeAircraft,
        strtoupper($registrationPlane),
        strtoupper($icaoDeparture),
        strtoupper($icaoArrival),
        $flightType,
        $flightMode,
        $weather,
        $passengers,
        $notes
    );
    //we show a success message
    $successMessage = "Vol enregistré avec succès!";
} catch (\Throwable $th) { //we show an error message
    $message = "Une erreur est survenue. Merci de réessayer";
}

```

#### 7.2.5.2 Modifier un vol

Cette fonctionnalité est accessible uniquement un utilisateur connecté. Les données du vol saisies par l'utilisateur écrasent celles déjà présentes dans la base de données si elles sont valides. Un message de succès ou d'erreur s'affiche en fonction du résultat.

```

if (empty($message)) {

    //we check with is the latest id in the database
    $allflight = FlightDAO::getAllFlightOrderById();
    try {

        //converting departure datetime to a good datetime
        $dateDeparture = explode("T", $dateDeparture);
        $dateDeparture = $dateDeparture[0] . " " . $dateDeparture[1];

        //converting arrival datetime to a good datetime
        $dateArrival = explode("T", $dateArrival);
        $dateArrival = $dateArrival[0] . " " . $dateArrival[1];

        //we edit the flight
        FlightDAO::editFlight(
            $flightId,
            $_SESSION['userID'],
            $role,
            strtoupper($flight),
            $dateDeparture,
            $dateArrival,
            $timeEngineOn,

```

```
        $timeEngineOff,  
        $typeAircraft,  
        strtoupper($registrationPlane),  
        strtoupper($icaoDeparture),  
        strtoupper($icaoArrival),  
        $flightType,  
        $flightMode,  
        $weather,  
        $passengers,  
        $notes  
    );  
    //we show a success message  
    $successMessage = "Vol modifié avec succès!";  
  
    //we get the new data to show to the user  
    $userFlight = FlightDAO::getFlightById($flightId);  
} catch (\Throwable $th) { //we show an error message  
    $message = "Une erreur est survenue. merci de réessayer";  
}
```

## 7.2.6 Groupes

### 7.2.6.1 Inviter un utilisateur au groupe

Cette fonctionnalité est seulement disponible si on est membre d'un groupe. Elle envoie une invitation à un utilisateur avec son email. L'invitation est seulement visible sur le site web, pas d'email envoyé, pas de notifications. Nous affichons un message de succès ou d'erreur en fonction du résultat de l'invitation.

```
//we check that the email field is not empty  
if (!empty($email)) {  
    //we get all the user data and try to invite him  
    $userData = UserDAO::getUserByEmail($email);  
    try {  
        if (!empty($userData)) { //if the user is existing we invite him and show a success message  
            GroupDAO::CreateInvite($idGroup, $userData['Id_User']);  
            $successMessage = "Utilisateur invité!";  
        } else { //if the user doesn't exists, we show an error message  
            $errorMessage = "Oops, une erreur est survenue. Vérifiez l'adresse du destinataire";  
        }  
    } catch (\Throwable $th) { //if the user is already in the group or is already invited, we show an error message  
        $errorMessage = "Oops, une erreur est survenue. Vérifiez l'adresse du destinataire";  
    }  
}
```

### 7.2.6.2 Créer un groupe

Cette fonctionnalité est disponible à tous les utilisateurs connectés. L'utilisateur nomme son groupe et clique sur le bouton « Créer ». Un message de succès s'affichera si le groupe a bien été créé. L'utilisateur sera automatiquement ajouté à son groupe fraîchement créé. Cette fonctionnalité regarde aussi que l'utilisateur ne saisisse pas un nom de plus de 100 caractères.

```
//we check that the name is not empty and not bigger than the maximum size in
the database
if (!empty($name) && strlen($name) <= 100) {

    //we create the group and get his id
    GroupDAO::createGroup($name);
    $lastID =GroupDAO::getLastGroupId();

    //we make the user join his newly created group
    GroupDAO::joinCreatedGroup($lastID[0]['Id_Group'], $_SESSION['userID']);

    //we show him a success message
    $message = "Groupe créer avec succès!";

}elseif(strlen($name) > 100){
    //we show an error message
    $errorMessage = "Nom de groupe trop long";
}
```

### 7.2.6.3 Liste des membres du groupe

Cette fonctionnalité est disponible à tous les membres du groupe concerné. Au menu du groupe, l'utilisateur clique sur « Liste des membres ». Un tableau les données de tous les membres avec pour chaque membre l'email et la date à laquelle il a rejoint le groupe s'affiche.

```
/**
 * function that show all group members
 *
 * @return string
 */
function showGroupMembers($idGroup)
{
    //we get all users of the selected group in a variable
    $groupMember = GroupDAO::getAllOfGroupUser($idGroup);

    //we write every user of the selected group
    foreach ($groupMember as $key => $value) {
        echo "<tr><td>" . $value['Txt_Email'] . "</td><td>" . $value['Dttm_Mem
bership'] . "</td></tr>";
    }
}
```

#### 7.2.6.4 Quitter un groupe

Cette fonctionnalité est disponible à tous les membres d'un groupe. En cliquant sur Mes groupes, une liste des groupes s'affiche avec chacun deux boutons. En cliquant sur la croix « X » rouge, l'utilisateur quitte le groupe et ses heures de vols disparaissent du carnet de vol du groupe. Un message de succès s'affiche si l'opération réussit et un message d'erreur s'affiche en cas d'erreur.

```
$leaveGroup = filter_input(INPUT_POST, 'delete', FILTER_SANITIZE_NUMBER_INT);

//we initialize empty messages
$errorMessage = "";
$successMessage = "";

//check if the user clicked on Leave group
if (!empty($leaveGroup)) {
    try { //we try to remove the user from the group
        GroupDAO::deletePendingInvite($leaveGroup, $_SESSION['userID']);
        $successMessage = "Vous avez quitter le groupe!";
    } catch (\Throwable $th) {
        //if there was an error
        $errorMessage = "Une erreur est survenue, veuillez réessayer";
    }
}
```

#### 7.2.6.5 Accepter ou refuser une invitation

Cette fonctionnalité est disponible à tous les utilisateurs connectés. Au menu du groupe, l'utilisateur clique sur « Mes invitations ». Un tableau apparaît alors en lui montrant toutes les invitations en attente ou un message lui disant qu'il n'a pas reçu d'invitations.

```
//check if the user clicked on any of these buttons
$accept = filter_input(INPUT_POST, "accept", FILTER_SANITIZE_STRING);
$deny = filter_input(INPUT_POST, "delete", FILTER_SANITIZE_STRING);

//we initialize the default error or success message
$successMessage = "";

//if he accepted an invite, we update the database
if ($accept) {
    GroupDAO::acceptPendingInvite($accept, $_SESSION['userID']);
    $successMessage = "Groupe rejoint avec succès";
}

//if he deny an invite, we remove it from the database
if ($deny) {
    GroupDAO::deletePendingInvite($deny, $_SESSION['userID']);
    $successMessage = "Invitation refusée avec succès";
}
```



```

/**
 * function that show all invites for the current user
 *
 * @return string
 */
function showInvites()
{
    //we get all invites in a variable
    $pendingInvites = GroupDAO::getAllPendingInvites();

    //we initialize a counter
    $countUserInvite = 0;
    //we check every invite of the database
    foreach ($pendingInvites as $key => $value) {
        //if an invite is for our user, we show it to him
        if ($value['Id_User'] == $_SESSION['userID']) {
            //we increment our counter
            $countUserInvite++;
            echo "<tr><td>Vous avez été invité à rejoindre le groupe " . GroupDAO::getGroupById($value['Id_Group'])['Nm_Group'] . " !</td><td><button class=\"btn btn-success\" type=\"submit\" value=\" " . $value['Id_Group'] . "\" name=\"accept\">>✓</button></td><td><button class=\"btn btn-danger\" type=\"submit\" value=\" " . $value['Id_Group'] . "\" name=\"delete\">>X</button></td></tr>";
        }
    }
    //if our user have 0 invites, we show him a message telling him to come back later to check if he got a new invite
    if ($countUserInvite == 0) {
        echo "<tr><td>Aucune invitation pour le moment... </td></tr><tr><td>Revenez plus tard</td></tr>";
    }
}

```

## 7.2.7 Téléchargement

### 7.2.7.1 Csv

L'utilisateur connecté a la possibilité de télécharger son carnet de vol au format CSV. Le CSV est généré en PHP et est directement téléchargé par l'utilisateur. Le CSV n'est pas stocké sur le serveur. L'utilisateur reste sur la page où il est au moment où il clique sur le bouton « Télécharger en CSV ». Dans le fichier se trouvent toutes les données du vol importantes en plus de l'adresse email du pilote, un total d'heures et une date de quand le document a été généré.

### 7.2.7.2 PDF

L'utilisateur connecté a la possibilité de télécharger son carnet de vol au format PDF. Le PDF est généré en PHP avec la classe « TCPDF ». Le PDF n'est pas stocké sur le serveur. L'utilisateur est

redirigé sur la page du PDF et à la possibilité de l'imprimer ou de le télécharger. Dans le fichier se trouve toutes les données du vol importantes en plus de l'adresse email du pilote, un total d'heures et une date de quand le document a été généré.

### 7.2.8 Ajout d'images

L'utilisateur connecté a la possibilité d'enregistrer les images de son vol. Les images sont stockées sur le serveur et le lien de ces images est stocké dans la base de donnée. L'utilisateur a la possibilité d'ajouter et de supprimer les images qu'il a importé sur le serveur à tout moment.

### 7.2.9 Suppression d'images

L'utilisateur connecté a la possibilité de supprimer les images de son vol enregistrée sur le serveur. La suppression est faite en PHP. Quand l'utilisateur clique sur la croix, on cache l'image et ajoute son id dans un « *input hidden* ». Dès que l'utilisateur clique sur modifier, on récupère les id de l'input et on les supprime si ils sont valides (voir la sécurité pour plus de détails).

## 7.3 Sécurité du carnet de vol

### 7.3.1 Carnet personnel

Le carnet personnel est sécurisé avec la session PHP. On ne prend que les vols de l'utilisateur actuellement connecté. Si aucun vol n'est trouvé, on affiche une ligne de « / ».

```
//we check if the user is already logged-in
if (!isset($_SESSION['userID'])) {
    header("Location: ./index.php?page=homepage");
    exit();
}

//we get all flight from the current user
$userFlight = FlightDAO::getAllUserFlightById($_SESSION['userID']);

/**
 * Show all flight of the current user
 *
 * @param array[mixed] $flight
 * @return void
 */
function showFlight($flight)
{
    //if there is at least 1 flight, we show it
    if (!empty($flight)) {
        foreach ($flight as $key => $value) {
            echo "<tr><td>" . $value['No_Flight'] . "</td><td>" . $value['Dttm_Departure'] . "</td><td>" . $value['Dttm_Arrival'] . "</td><td>" .
```

```

        $value['Tm_Engine_On'] . "</td><td>" . $value['Tm_Engine_Off']
        . "</td><td>" . $value['Nm_Plane'] . "</td><td>" . $value['No_Plane'] . "</td>
<td>" .

        $value['Cd_ICAO_Departure'] . "</td><td>" . $value['Cd_ICAO_Ar
rival'] . "</td><td>" . $value['Cd_Flight_Type'] . "</td><td>" . $value['Cd_Fl
ight_Mode'] .

        "</td><td>" . $value['Cd_Role'] . "</td><td><a href='?page=log
bookDetail&id=" . $value['Id_Flight'] . "'>Détails</a></td><td><a href='?page=
logbookEdit&id=" . $value['Id_Flight'] . "'>Editer le vol</a></td></tr>";
    }
} else { //else we show "/"
    echo "<tr><td></td><td></td><td></td><td></td><td></td><td></td>
<td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td>
<td></td><td></td></tr>";
}
}
}

```

### 7.3.2 Carnet de groupe

Le carnet de groupe fonctionne comme le croquis ci-dessous :

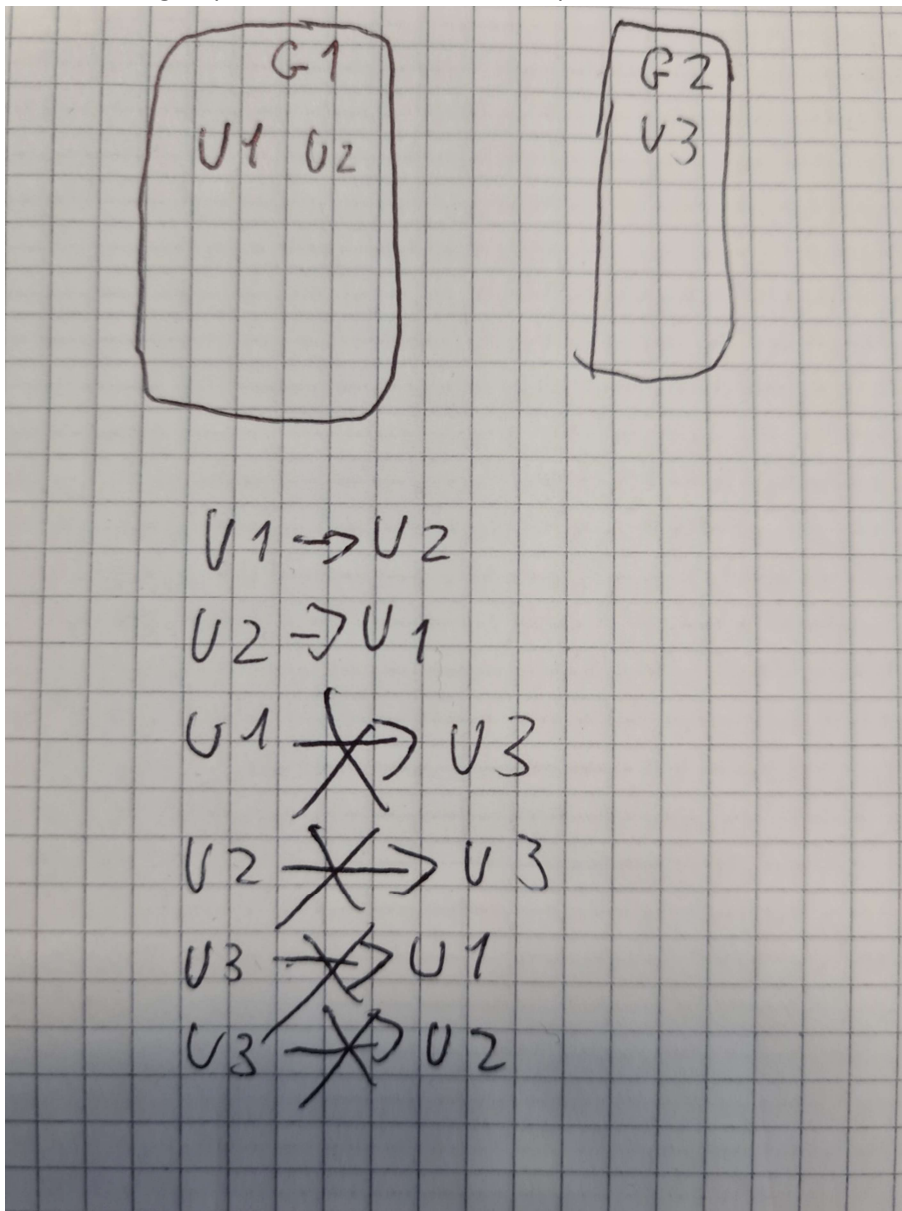


Fig. droits sur les carnets de groupes

G1 = groupe 1, G2 = Groupe 2, U1 = utilisateur1, U2 = utilisateur2 et U3 = utilisateur3

Le carnet de groupe est sécurisé via l'URL. Imaginons que l'utilisateur 1 à enregistrer un vol avec l'id 1. L'URL du vol vas être: « index.php?page=logbookDetail&id=1 ». Si l'utilisateur 2 essaie cette URL, il va avoir accès aux données du vol. Mais si l'utilisateur 3 décide de modifier l'url d'un de ses vol pour mettre l'id 1 du vol de l'utilisateur 1, le site va lui afficher un 404.

### 7.3.3 La suppression d'images

La suppression d'images est sécurisée de manière à empêcher un utilisateur de modifier l'id de supprimer l'image d'un autre utilisateur en modifiant l'id de son image. Le fonctionnement est le suivant : On vérifie que l'id de l'image à supprimer appartient à un vol de l'utilisateur connecté. Si c'est le cas, on supprime l'image, sinon, on ne supprime pas l'image

## 8 Plannification

Sur ce diagramme, vous pouvez trouver ma planification prévue pour les tâches du projet, ainsi que la façon dont la réalisation des tâches s'est déroulée.

### Planning

TITRE DU PROJET		Flight Club	NOM DE L'ENTREPRISE		CFPT - Informatique
ELEVE		Golay Brian	DATE		20/05/21

NUMÉRO	TITRE DE LA TÂCHE/USER STORY	DURÉE (H)	SEMAINE 1					SEMAINE 2					SEMAINE 3					Total
			J0	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J12	J13	J14	
			lu. 03	ma. 04	me. 05	je. 06	ve. 07	lu. 10	ma. 11	me. 12	je. 13	ve. 14	lu. 17	ma. 18	me. 19	je. 20	ve. 21	
<b>1</b>	<b>Plannification et mise en route du projet</b>	<b>10</b>																<b>10</b>
	Lecture de l'énoncé	1																
	Rédaction du product backlog	2																
	Planification des tâches	2																
	Rédaction des scénarios de tests	2																
	Utilisation d'un dépôt Git	1																
	Configuration de la base de données	2																
<b>2</b>	<b>Implémentation</b>	<b>35</b>																<b>35</b>
	Gestion des utilisateurs	4																
	Gestion des groupes	5																
	Affichage de la landing page	2																
	Affichage du carnet de vol	2																
	Gestion du carnet de vol	3																
	Gestion du Profile	4																
	Téléchargement du carnet	4																
	Gestion d'attachement	6																
	Finitions	10																
<b>3</b>	<b>Documentation et contrôle</b>	<b>37</b>																<b>37</b>
	Tests en profondeur et corrections de bugs	8																
	Rédaction de la documentation	24																
	Tenue du journal de bord	5																
		<b>82</b>																<b>82</b>

● Planification prévue ● Planification effective

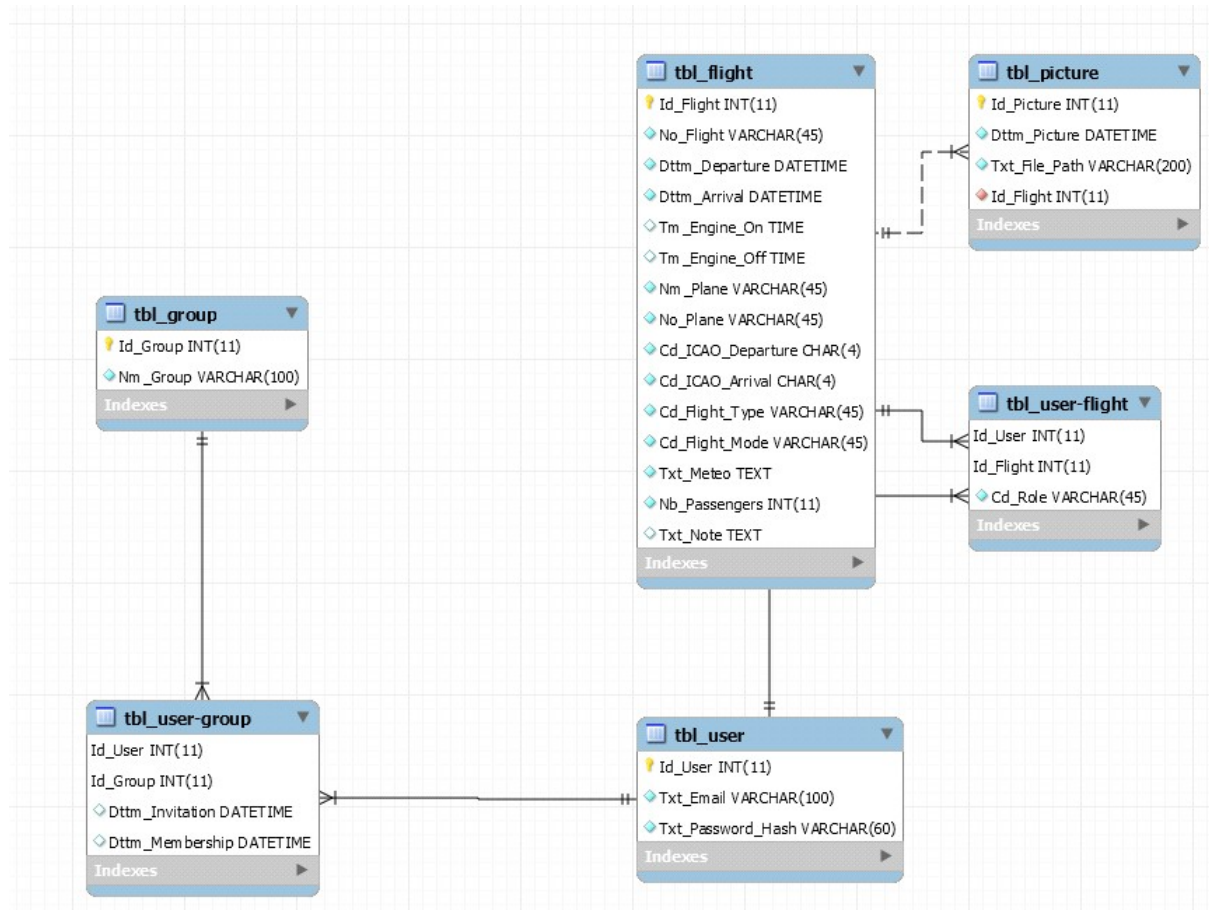
## 9 Généralités concernant l'implémentation

## 9.1 Base de données

J'utilise une base de donnée MySQL 5.7.17 pour stocker toutes les données du site. J'ai utilisé InnoDB comme moteur de stockage et utf8\_general\_ci comme interclassement.

Le modèle de données à utiliser était fourni dans l'énoncé (Il a légèrement été modifié comme discuté en classe avec M.Mathieu et expliqué par e-mail).

### 9.1.1 Le modèle de données final



J'ai dû créer un script SQL permettant de créer les tables nécessaires, lors du premier jour du TPI. Ce dernier se trouve dans le répertoire suivant : /sql/ il se nomme structure.sql

## 9.1.2 Dictionnaire de données

### 9.1.2.1 Table Tbl\_Flight

Colonne	Type	Null	Valeur par défaut	Commentaire
Id_Flight	Int	Non		Clef primaire
No_Flight	Varchar(45)	Non		
Dttm_Departure	DateTime	Non		
Dttm_Arrival	DateTime	Non		
Tm_Engine_On	Time	Oui		
Tm_Engine_Off	Time	Oui		
Nm_Plane	Varchar(45)	Non		

No_Plane	Varchar(45)	Non		
Cd_ICAO_Departure	Char(4)	Non		
Cd_ICAO_Arrival	Char(4)	Non		
Cd_Flight_Type	Varchar(45)	Non		
Cd_Flight_Mode	Varchar(45)	Non		
Txt_Meteo	Text(512)	Non		
Nb_Passengers	Int	Non		
Txt_Note	Text(1024)	Oui		

#### 9.1.2.2 Table Tbl\_Picture

Colonne	Type	Null	Valeur par défaut	Commentaire
Id_Picture	Int	Non		Clef primaire
Dttm_Picture	DateTime	Non		
Txt_File_Path	Text(200)	Non		
Id_Flight	Int	Non		Clef étrangère

#### 9.1.2.3 Table Tbl\_User-Flight

Colonne	Type	Null	Valeur par défaut	Commentaire
Id_User	Int	Non		Clef étrangère
Id_Flight	Int	Non		Clef étrangère
Cd_Role	Varchar(45)	Non		

#### 9.1.2.4 Table Tbl\_User

Colonne	Type	Null	Valeur par défaut	Commentaire
Id_User	Int	Non		Clef primaire
Txt_Email	Varchar(100)	Non		
Txt_Password_Hash	Varchar(60)	Non		

#### 9.1.2.5 Table Tbl\_User-Group

Colonne	Type	Null	Valeur par défaut	Commentaire
Id_User	Int	Non		Clef étrangère
Id_Group	Int	Non		Clef étrangère

Dttm_Invitation	DateTime	Oui		
Dttm_Membership	DateTime	Oui		

#### 9.1.2.6 Table Tbl\_Group

Colonne	Type	Null	Valeur par défaut	Commentaire
Id_Group	Int	Non		Clef primaire
Nm_Group	Varchar(100)	Non		

## 10 Librairies et outils externes

### 10.1 Git

Le logiciel de gestion de versions Git a été utilisé tout au long de la création du projet afin de garder un historique des modifications effectuées, de synchroniser en permanence les dernières versions du code source et d'avoir une sauvegarde externe en cas de défaillance technique ou de corruptions de données. Un dépôt distant contenant le code source du projet est disponible sur GitHub (droits d'accès nécessaires) : <https://github.com/BRIANGL/Flight-Club>



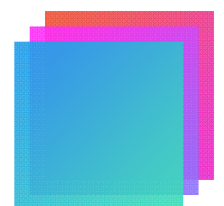
### 10.2 Bootstrap

Bootstrap est un framework open-source que j'ai utilisé durant mon projet afin d'obtenir un design agréable, car il contient des classes pré-faites pour les formulaires, les boutons et d'autres composants d'interface. <https://getbootstrap.com>



### 10.3 Bootstrap Studio

Bootstrap studio est un outil payant que j'ai utilisé durant mon projet afin de me simplifier le design du site web. Il permet de créer un design utilisables et modifiable a souhait rapidement. Il permet aussi d'exporter le travail en un projet html, css et JavaScript.



### 10.4 Flaticon

Un moteur de recherche utile pour trouver des icônes libres de droits

### 10.5 Expression régulière

Les expressions régulières, plus communément appelées regex sont utilisées en informatique afin de faciliter les procédures de contrôles d'entrées, de lecture, d'analyse ou encore de modification. Dans



le cadre de mon application, je les utilise dans le simple but de vérifier la force d'un mot de passe lors de l'enregistrement (une majuscule, une minuscule, un chiffre et un caractère spécial).

<https://regex101.com/>

### 10.6 Katalon Recorder

C'est une extension de navigateur, visant à aider les développeurs pour les tests et le débogage des applications. C'est une extension qui m'a aidé recommandé par mon formateur, et qui m'a servi tout au long de mon travail, afin de vérifier à chaque étape, qu'il n'y avait pas de régression.



### 10.7 Trello

Trello est un outil de planification très utile pour savoir rapidement ce que l'on doit faire dans notre programme ou ce que nous avons déjà fait.



### 10.8 TCPDF

TCPDF est un outil pour générer des PDF en PHP facilement. C'est une librairie très utile et qui peut aussi servir à générer des codes QR ou codes-barres

### 10.9 Figma

Figma est un outil pour faire des schéma navigation simples

## 11 Plan de tests et tests

### 11.1 Périmètre des tests

Pour le projet Flight Club, j'ai choisi d'écrire des protocoles de test pour les actions qu'un utilisateur  $\lambda$  (lambda) utilisant un navigateur moderne et répandu pourrait raisonnablement effectuer, pour ce qui est des questions que l'utilisateur pourrait se poser, elles sont disponibles sur la page `faq.php` et présentent certaines contraintes actuelles de mon implémentation.

### 11.2 Équipement de test

#### 11.2.1 Environnement

Lors du développement du projet, j'ai utilisé l'environnement suivant pour effectuer les tests : Google Chrome version 90.0.4430.93 (64 bits) sur Windows 10, version 10.0.19041 build 19041 Côté serveur, apache 2.4.25 a été utilisé avec PHP 7.1.3 et MySQL 5.7.17. Le système utilisé comme serveur est la machine Windows citée ci-dessus.

### 11.3 Scénarios de test

Lors du premier jour du TPI, j'ai rédigé les scénarios de tests suivants qui m'ont permis par la suite de m'assurer que les besoins exprimés dans l'énoncé sont remplis. Les scénarios de tests sont précis dans le but d'être facilement reproductibles par une autre personne si nécessaire.

Nom	1.1 Affichage de la <i>home</i> page
User story	S3 : affichage de la home page
Situation	Étant donné que je suis un utilisateur non connecté Quand je me rends sur le site Alors une home page s'affiche. Cette page me présente brièvement le projet et contient un bouton « Se connecter » qui permet de se connecter au site.
Résultats obtenus	Je me retrouve redirigé sur une page de connexion me demandant de remplir des champs de types email et password avec mes données de connexion.
Statut	✓ Ok

Nom	2.1 Connexion (compte existant)
User story	S7 : Connexion à son compte
Situation	Étant donné que je suis un utilisateur non connecté, qui possède déjà un compte sur le site. Quand je clique sur le bouton « Se connecter » sur la page de login et que je rentre les bons identifiants. Alors, je suis redirigé vers ma page profil, en étant connecté à mon compte existant.
Résultats obtenus	Je clique sur Se connecter. Je suis redirigé vers une page d'authentification, je rentre mes identifiants, et me retrouve sur la « <i>home page</i> », et la barre de navigation à changer.
Statut	✓ Ok

Nom	2.2 Connexion (erreur)
User story	S7 : Connexion à son compte
Situation	Étant donné que je suis un utilisateur non connecté, qui possède déjà un compte sur le site. Quand je clique sur le bouton « Se connecter » sur la page de login et que je rentre les mauvais identifiants. Alors, je reçois un message d'erreur.
Résultats obtenus	Étant donné que je suis un utilisateur non connecté, qui possède déjà un compte sur le site. Quand je clique sur le bouton « Se connecter » sur la page de login et que je rentre

	les mauvais identifiants. Alors, je reçois un message d'erreur.
Statut	✓ Ok

Nom	3.1 Inscription
User story	S6 : Création de compte
Situation	Étant un utilisateur anonyme, ne possédant pas encore de compte, je souhaite me créer un compte. Quand je suis sur la page inscription, il me suffit de remplir les champs requis.
Résultats obtenus	D'une fois tous les champs remplis, j'appuie sur le bouton « S'enregistrer » et me fait rediriger à la page de connexion, sans être connecté.
Statut	✓ Ok

Nom	3.2 Inscription - erreur de saisie
User story	S6 : Création de compte
Situation	Étant un utilisateur anonyme, ne possédant pas encore de compte, je souhaite me créer un compte. Quand je suis sur la page inscription, si je mets une adresse mail trop longue lorsque je clique sur le bouton « S'enregistrer », je reçois un message d'erreur me demandant de vérifier mes entrées.
Résultats obtenus	Je vois s'afficher une balise en rouge m'indiquant de vérifier mes entrées.
Statut	✓ Ok

Nom	4.1 Importation de pièce jointe
User story	S4 : Enregistrements des vols
Situation	Étant donné que je suis un utilisateur connecté, qui possède déjà un compte sur le site. Je peux importer une photos de mon vols quand j'enregistre un de mes vols.
Résultats obtenus	Une fois les images ajoutées, quand je clique sur enregistrer le vol, un message de succès s'affiche et les images sont visibles dans les détails des vols
Statut	✓ Ok

Nom	4.2 Erreur Importation de pièce jointe
User story	S4 : Enregistrements des vols
Situation	Étant donné que je suis un utilisateur connecté, qui possède déjà un compte sur le site. J'enregistre une image d'un type pas accepté ou supérieur à 5Mo lors de l'enregistrement d'un vol et un message d'erreur apparait.

Résultats obtenus	Je vois s'afficher une balise en rouge m'indiquant que l'importation n'a pas été finie.
Statut	✓ Ok

Nom	5.1 Enregistrement de vol – Pas d'erreur de saisie
User story	S4 : Enregistrements des vols
Situation	Étant donné que je suis un utilisateur connecté, qui possède déjà un compte sur le site. J'enregistre mon vol sur le site avec succès. Un message de succès apparaît.
Résultats obtenus	Une fois les champs remplis, je clique sur le bouton « Enregistrer le vol » et un message de succès apparaît.
Statut	✓ Ok

Nom	5.2 Enregistrement de vol – Erreur de saisie
User story	S4 : Enregistrements des vols
Situation	Étant donné que je suis un utilisateur connecté, qui possède déjà un compte sur le site. J'enregistre mon vol avec des informations invalides. Un message d'erreur apparaît.
Résultats obtenus	Je vois s'afficher une balise en rouge m'indiquant de vérifier mes entrées.
Statut	✓ Ok

Nom	6.1 Modification de vol – Pas d'erreur de saisie
User story	S8 : Modification des vols
Situation	Étant donné que je suis un utilisateur connecté, qui possède déjà un compte sur le site. J'ai enregistré un vol erroné. Je le corrige et enregistre la modification. Un message de succès apparaît.
Résultats obtenus	Une fois les champs remplis, je clique sur le bouton « Modifier le vol » et un message de succès apparaît.
Statut	✓ Ok

Nom	6.2 Modification de vol – Erreur de saisie
User story	S8 : Modification des vols
Situation	Étant donné que je suis un utilisateur connecté, qui possède déjà un compte sur le site. J'ai enregistré un vol erroné. Je le corrige avec des données invalides et enregistre la modification. Un message d'erreur apparaît.
Résultats obtenus	Je vois s'afficher une balise en rouge m'indiquant de vérifier mes entrées.

Statut	✓ Ok
--------	------

Nom	7.1 FAQ
User story	S16 : Accès à la FAQ
Situation	Lorsqu'un utilisateur veut en savoir plus sur le site, il lui suffit de cliquer sur le lien menant à la FAQ, page sur laquelle il apprendra quelques fonctionnalités et limites de l'implémentation actuelle du site.
Résultats obtenus	Je clique sur FAQ, et la page s'affiche
Statut	✓ Ok

Nom	8.1 Téléchargement en CSV
User story	S5 : Téléchargement du carnet de vol
Situation	Étant donné que je suis un utilisateur connecté, qui possède déjà un compte sur le site. Je clique sur le bouton télécharger en .CSV et le fichier se télécharge dans le format sélectionnés.
Résultats obtenus	Je clique sur le bouton « Télécharger en CSV » et le téléchargement commence
Statut	✓ Ok

Nom	8.2 Téléchargement en PDF
User story	S5 : Téléchargement du carnet de vol
Situation	Étant donné que je suis un utilisateur connecté, qui possède déjà un compte sur le site. Je clique sur le bouton télécharger en .PDF et le fichier se télécharge dans le format sélectionnés.
Résultats obtenus	Je clique sur le bouton « Télécharger en PDF » et l'affichage du PDF s'affiche et me permet de le télécharger
Statut	✓ Ok

Nom	9.1 Modification du mot de passe - Succès
User story	S9 : Modification de mot de passe
Situation	Étant donné que je suis un utilisateur connecté, qui possède déjà un compte sur le site. Je remplis les champs de modification de mot de passe et je clique sur modifier mon mot de passe. Un message de succès apparaît.
Résultats obtenus	Je remplis les champs demandé et clique sur « Modifier ». Un message de succès s'affiche
Statut	✓ Ok

Nom	9.2 Modification du mot de passe - Erreur
User story	S9 : Modification de mot de passe

Situation	Étant donné que je suis un utilisateur connecté, qui possède déjà un compte sur le site. Je remplis les champs de modification de mot de passe qui ne passe pas les tests de sécurité minimum et je clique sur modifier mon mot de passe. Un message d'erreur apparaît.
Résultats obtenus	Je remplis les champs demandés avec des données incorrectes et clique sur « Modifier ». Un message d'erreur s'affiche
Statut	✓ Ok

Nom	10.1 Création de groupe - Succès
User story	S10 : Créer un groupe
Situation	Étant donné que je suis un utilisateur connecté, qui possède déjà un compte sur le site. Je remplis les champs de création de groupe et je clique sur créer un groupe. Un message de succès apparaît.
Résultats obtenus	Je remplis le champ du nom du groupe et clique sur le bouton « Créer ». Un message de succès s'affiche
Statut	✓ Ok

Nom	10.2 Création de groupe - Erreur
User story	S10 : Créer un groupe
Situation	Étant donné que je suis un utilisateur connecté, qui possède déjà un compte sur le site. Je remplis les champs de création de groupe avec des informations invalides et je clique sur créer un groupe. Un message d'erreur apparaît.
Résultats obtenus	Je remplis le champ du nom du groupe avec des données incorrectes et clique sur le bouton « Créer ». Un message d'erreur s'affiche
Statut	✓ Ok

Nom	11.1 Inviter des utilisateur à mon groupe - Succès
User story	S11 : Inviter des utilisateurs à mon groupe
Situation	Étant donné que je suis un utilisateur connecté, qui possède déjà un compte sur le site. Je remplis les champs d'invitation de groupe et je clique sur inviter au groupe. Un message de succès apparaît.
Résultats obtenus	Je remplis le champ de l'email de l'utilisateur à inviter et clique sur le bouton « Inviter un membre ». Un message de succès s'affiche
Statut	✓ Ok

Nom	11.2 Inviter des utilisateur à mon groupe - Erreur
User story	S11 : Inviter des utilisateurs à mon groupe
Situation	Étant donné que je suis un utilisateur connecté, qui possède déjà un compte sur le site. Je remplis les champs d'invitation de groupe avec des données invalides et je clique sur inviter au groupe. Un message d'erreur apparaît.
Résultats obtenus	Je remplis le champ de l'email de l'utilisateur à inviter avec un email invalide et clique sur le bouton « Inviter un membre ». Un message d'erreur s'affiche
Statut	✓ Ok

Nom	12.1 Accepter une invitation de groupe
User story	S12 : Accepter une invitation à un groupe
Situation	Étant donné que je suis un utilisateur connecté, qui possède déjà un compte sur le site. Je clique sur accepter l'invitation et un message de succès apparaît.
Résultats obtenus	Je clique sur le bouton « ✓ » et un message de succès s'affiche
Statut	✓ Ok

Nom	13.1 Refuser une invitation de groupe
User story	S13 : Refuser une invitation à un groupe
Situation	Étant donné que je suis un utilisateur connecté, qui possède déjà un compte sur le site. Je clique sur refuser l'invitation et un message de succès apparaît.
Résultats obtenus	Je clique sur le bouton « X » et un message de succès s'affiche
Statut	✓ Ok

Nom	14.1 Affichage des vols dans les groupes auxquels j'appartiens
User story	S14 : Affichage des vols dans les groupes
Situation	Étant donné que je suis un utilisateur connecté, qui possède déjà un compte sur le site. Je clique sur le carnet de vol du groupe et tous les vols des utilisateurs du groupe apparaissent.
Résultats obtenus	Je clique sur le bouton « Carnet de vol du groupe » et les vols du groupe s'affichent
Statut	✓ Ok

Nom	15.1 Quitter un groupe
User story	S15 : Quitter un groupe

Situation	Étant donné que je suis un utilisateur connecté, qui possède déjà un compte sur le site. Je clique sur le bouton « X » le groupe et un message de succès apparaît. Je n'ai plus accès aux vols de ce groupe
Résultats obtenus	Je clique sur le bouton « X » et le groupe disparaît et un message de succès s'affiche
Statut	✓ Ok

N° test	J0	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10
1.1	X	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
2.1	X	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
2.2	X	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
3.1	X	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
4.1	X	X	X	X	X	✓	✓	✓	✓	✓	✓
4.2	X	X	X	X	X	✓	✓	✓	✓	✓	✓
5.1	X	X	X	✓	✓	✓	✓	✓	✓	✓	✓
5.2	X	X	X	✓	✓	✓	✓	✓	✓	✓	✓
6.1	X	X	X	✓	✓	✓	✓	✓	✓	✓	✓
6.2	X	X	X	✓	✓	✓	✓	✓	✓	✓	✓
7.1	X	X	X	X	X	X	✓	✓	✓	✓	✓
8.1	X	X	X	X	X	✓	✓	✓	✓	✓	✓
8.2	X	X	X	X	✓	✓	✓	✓	✓	✓	✓
9.1	X	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
9.2	X	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
10.1	X	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
10.2	X	X	X	X	✓	✓	✓	✓	✓	✓	✓
11.1	X	X	✓	✓	✓	✓	✓	✓	✓	✓	✓
11.2	X	X	✓	✓	✓	✓	✓	✓	✓	✓	✓
12.1	X	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
13.1	X	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
14.1	X	X	✓	✓	✓	✓	✓	✓	✓	✓	✓
15.1	X	X	✓	✓	✓	✓	✓	✓	✓	✓	✓

## 12 Conclusion

### 12.1 Difficultés rencontrées

Je n'ai pas eu de réel difficulté majeure, les petits problèmes qui ont pris plus de temps que prévus étaient des erreurs d'inattention ou oublis comme lors que je faisais une requête SQL pour un champ et ou j'ai perdu du temps car j'essayais avec le « = » alors que c'était « is » qu'il fallait mettre. Dans l'ensemble, tout s'est bien déroulé.



## 12.2 Améliorations possibles

### 12.2.1 Architecture technique

La structure actuelle de l'application fonctionne bien : la structure du projet est déjà organisée de manière professionnelle, en un découpage permettant de séparer la partie utilisateur de la partie serveur.

### 12.2.2 Fonctionnalités

Sur le court terme, les améliorations suivantes concernant des fonctionnalités déjà existantes pourraient être réalisées :

- Ajout de rôles avec différentes permissions.
- Ajout de JavaScript pour des requêtes asynchrones ex : L'utilisateur entre LSMP, on met dans un champ à côté Payerne.
- Ajout de sécurité (Captcha, .htaccess, etc ...).
- Un système de pagination pour le carnet de vol et la liste des utilisateurs dans un groupe.
- Un système de tri des vols dans le carnet de vol ex : ICAO de départ alphabétique, date de départ, type d'appareil, immatriculation, etc ...
- Un total d'heure par appareil différent.

## 12.3 Bilan personnel

Ce projet réalisé sur une durée de 11 jours, m'a permis de confirmer mes connaissances professionnelles, ainsi que la mise en application d'un planning de travail. J'ai appris rapidement que faire la documentation tout au long du projet est plus agréable que la faire à la fin. Je suis content d'avoir eu la chance d'avoir ce projet car il correspond à ma passion et pourra me servir par la suite. De plus que ce projet est une excellente base pour tout type de future projet web. Il manque un peu de JavaScript et d'Ajax mais je suis satisfait de mon travail bien qu'il m'ait été imposé, et non choisi. Je pense que si je devais refaire ce projet demain, la seule chose qui changerait serait le planning. Je répartirais mieux la documentation sur l'ensemble du temps et ferais attention aux jours fériés et obligations civiles. Je suis content d'avoir des camarades et professeurs qui m'ont aidé que ce soit pour l'orthographe ou les fautes toutes simples dans le code.

## 12.4 Remerciements

Je tiens à remercier :

- M.Mathieu pour son accompagnement durant mon TPI.
- M.Zanardi pour son aide et son avis sur quelques points techniques.
- M.Garcia pour ses documents sur la documentation.
- Yvelin Ray pour la correction de mon orthographe et les tests de mon programme.

- Costantino Volta pour son astuce pour le footer qui reste en bas de la page.

## 13 Annexes

### 13.1 Bibliographie

Lors du déroulement de mon projet, j'ai utilisé les ressources suivantes pour obtenir de l'aide technique :

- Documentation officielle de Bootstrap 5 : <https://getbootstrap.com/docs/5.0/getting-started/introduction/Glossaire>
- La documentation officielle de PHP : <https://www.php.net/docs.php>
- Le forum de stackoverflow : <https://stackoverflow.com/>

## 14 Glossaire

### 14.1 Termes techniques

#### 14.1.1 User stories (ou récit utilisateur)

Description d'une fonctionnalité utilisée dans la méthodologie agile, qui se présente sous cette forme: En tant que <qui>, je veux <quoi> afin de <pourquoi>.

#### 14.1.2 Importation (Anglais : upload)

Fait de pouvoir importer des fichiers depuis son ordinateur.

#### 14.1.3 Carnet de produit (Anglais: product backlog)

Ensemble des besoins recueillis pour créer le produit.

#### 14.1.4 Formulaire GET

Un formulaire GET, permet de recevoir les paramètres depuis l'URL.

#### 14.1.5 Input hidden

Un champ caché à l'utilisateur.

#### 14.1.6 Framework

Ensemble important de briques logicielles sur lequel on peut se baser pour créer un logiciel informatique. Exemple de frameworks : Laravel (framework PHP), Vue.js (framework JavaScript front-end), Bootstrap (framework CSS JavaScript).

#### 14.1.7 Sticky

Un champ collant. Un champ que lors que l'on rafraîchit la page, les données ne sont pas effacées

#### 14.1.8 Hash

Un Hash est le résultat d'une opération mathématique pour cacher la chaîne de caractère original en une chaîne incompréhensible pour un utilisateur et où l'on ne peut pas revenir en arrière dans la méthode. Ex : abc -> fuihekhsh48h2h4iu1h23, fuihekhsh48h2h4iu1h23 -X> abc.

#### 14.1.9 Salt

Un salt sécurise un Hash. Il n'est pas obligatoire mais fortement conseiller. Un salt est une chaîne de caractère qu'on ajoute à un hash pour brouiller les pistes pour un éventuel attaquant qui essaierais de retrouver la chaîne de caractère originelle du hash.

## 15 Résumé du TPI

### Flight Club

Situation de départ : Pour valider la formation d'Informaticien-ne CFC, les apprentis doivent effectuer un « Travail pratique individuel » (TPI) d'une durée de 88 heures (11 jours), sur la base d'un énoncé imposé. Le projet qui m'a été attribué est une application web permettant à ses utilisateurs de pouvoir enregistrer et partager leurs vols avec d'autres personnes étant dans le même groupe. Les utilisateurs doivent pour pouvoir se connecter sur le site, enregistrer leurs vols et les sauvegarder au format PDF ou CSV à tout moment. Ils doivent également avoir la possibilité de créer, rejoindre et quitter des groupes de plus qu'inviter des membres au groupe.

Mise en œuvre : Le projet a été réalisé en employant PHP et MySQL côté serveur, et du HTML CSS et JavaScript. L'application prend la forme d'une page unique. En outre, la méthodologie dite de planification en 6 étapes a été utilisée tout au long du projet pour organiser le travail. Les différents points du cahier des charges ont été transformés en user stories permettant d'exprimer le besoin du point de vue des utilisateurs finaux, et un protocole de tests a été rédigé pour m'assurer de la conformité de l'implémentation par rapport à l'énoncé. Un journal de bord a été quotidiennement tenu à jour pour détailler le déroulement du projet, lister les imprévus et problèmes auxquels j'ai dû faire face, et expliquer les différents choix techniques. Pour les permissions, je suis parti du principe que l'utilisateur actuel est le propriétaire de ses vols. Donc il a le droit de les consulter et les modifier. Il a également le droit de consulter les vols de ses camarades de groupes mais pas de les modifier.

Résultats : Les utilisateurs finaux ont à leur disposition une application web simple d'utilisation, leur permettant d'enregistrer et partager leurs vols. L'intégralité des spécifications du cahier des charges a pu être implémentée.